ELSEVIER

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Environmental Sciences 12 (2012) 802 – 809

Procedia Environmental Sciences

2011 International Conference on Environmental Science and Engineering

(ICESE 2011)

Policy Driven and Agent Based Geospatial Information Services Composition*

Bin Xie[1,a], Dengrong Zhang[1,b], Xiang Li[2,c], Yang Wu[3,d]

[1]Institute of Remote Sensing and Earth Sciences, Hangzhou Normal University
Hangzhou, Zhejiang Province, China
[2]Zhejiang Institute of Geological Survey, Hangzhou, Zhejiang Province, China
[3]Institute of Spatial Information Techniques, Zhejiang University
Hangzhou, Zhejiang Province, China
[a]xiebinbingdiao@gmail.com, [b]zju_rs@126.com, [c]kyo.lixiang@163.com, [d]mikel.wuyang@gmail.com

Abstract

Nowadays, regarding to the initiative discovery and flexible management of distributed geospatial information services, no reliable and well developed methods or products are available due to the complexity of the web environment. By integrating semantic ontology, policy and software agent technologies, this paper proposed representation method of services collaborative management policy and policy driven self-management mechanism of geospatial agent. On this basis, multi-geospatial agents based collaborative service framework is provided to promote intelligent expression and flexible management of geospatial information service composition, and to reduce human interventions.

*Keywords :*Agent; policy driven; geospatial information services; service composition.

1.Introduction

In the opening, heterogeneous, distributed web environment, the geospatial information service's discovery, binding and invocation are based on OGC standards and web service interface. But this way

* This work is supported by National High Technology Research and Development Program (863) of China (Grant #2009AA12Z212) to B. Xie and D. Zhang.

 doi:10.1016/j.proenv.2012.01.351

has lack of intelligent means to achieve service automatic discovery, invocation and composition. Software agent, which has autonomy, initiative and social characteristics [1-3], can be used to implement automatic consultation and collaboration for geospatial information services. However, because of uncertain individual behaviors of a single agent, user's requirements or objects are necessary to dynamically drive the establishment of collaborative relationships among agents. In view of this consideration, policy-based management [4-11] is introduced to express the requirements or objects of users. The policy is deployed to agent for driving the collaborative behavior of its internal components without changing its software coding. Therefore, this paper combines management policy and software agent technologies to provide loosely coupled dynamic collaborative mechanism for geospatial information services composition.

2.Expression Of Policy

At present, there are four main policy expression specifications, including KAoS [5], Rei [12], Ponder [13] and PRAL (Policy Representation and Assignment Language) [14]. Among them, ontology is introduced into KAoS to define related concepts for policy description. Besides, negotiation policies between agents can be well expressed by PRAL. Therefore, this paper takes full advantages of KAoS and PRAL to describe management policies for multi-geospatial agents based geospatial information services composition. Here, the management policy is used to restrict behaviors of geospatial agent, such as discovery, collaboration and monitoring of geospatial information services.

Formally, the policy can be defined as the 8-tupe: Policy = {ID, Type, Action, Subject, Target, Trigger, Priority, Update}.

ID: denotes the policy identity. [typedef: string].

Type: denotes the policy type, including three types: authorization, obligation and negotiation. [typedef: PolicyType].

Subject: denotes the owner or executor of the policy. [typedef: Role].

Target : denotes the performing object (such as Geospatial Agent) of policy "Subject". [typedef: GeospatialAgent].

Action: denotes the proposed processes and actions, including operations and rules. The "Target" (such as geospatial agent) takes the behaviors required by "Subject". [typedef: PolicyOperation, PolicyRule].

Trigger: denotes the toggle condition of policy. The policy will not be started until the conditions are satisfied. [typedef: ConditionExpression].

Priority: denotes the policy performing priority. The policies conflicts appear when "Target" triggers multiple policies at the same time. Therefore, the policies are executed sequentially according to the priority. [typedef: int].

Update: denotes the updated date of policy (timestamp). [typedef: date]

Based on the above formal specification, a sample policy of WCS publication expressed by OWL is shown below:

```
<owl: Class rdf: ID = "WCS">
   <rdfs: subClassOf rdf: resource = "GeographicService"/>
   <rdfs: subClassOf>
      <owl: Restriction>
         <owl: onProperty rdf: resource = "#hasState"/>
         <owl: toClass rdf: resource = "#RegisteringIntoUDDI"/>
      </owl: Restriction>
   </rdfs: subClassOf>
</owl: Class>
<owl: Class rdf: ID = "Policy_RegisterAction">
   <rdfs: subClassOf rdf: resource = "#ServiceProfileRegisterAction"/>
   <rdfs: subClassOf>
      <owl: Restriction>
         <owl: onProperty rdf: resource = "#hasSubject"/>
         <owl: toClass rdf: resource = "#CSW"/>
      </owl: Restriction>
   </rdfs: subClassOf>
   <rdfs: subClassOf>
      <owl: Restriction>
         <owl: onProperty rdf: resource = "#hasObject"/>
         <owl: toClass rdf: resource = "#WCS"/>
      </owl: Restriction>
   </rdfs: subClassOf>
</owl: Class>
<policy: AuthorizationPolicy rdf: ID = "Policy_Register">
   <policy: hasAction rdf: resource = "# Policy_RegisterAction"/>
   <policy: hasSiteOfEnforcement rdf: resource = "#UDDI"/>
   <policy: hasSubjec rdf: resource = “#CSW”/>
   <policy: hasTarget rdf: resource = “#WCS”/>
   <policy: hasPriority>3</policy: hasPriority>
   <policy: hasUpdateTimeStamp>2011-01-15 12:00:00</policy: hasUpdateTimeStam>
</policy: AuthorizationPolicy>
```

3.Policy driven multi-geospatial agents collaboration

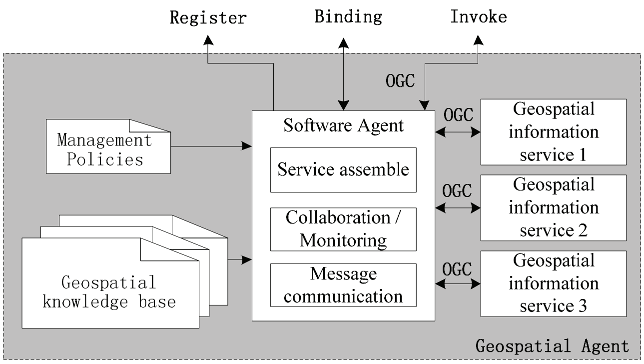

Fig. 1 Framework of geospatial agent

3.1.Framework of Geospatial Agent

The core component of geospatial agent is the software agent integrating with geospatial information services. The framework of geospatial agent is shown in Fig. 1.

Software agent including message communication module, collaborate module and service assemble module is the basic infrastructure of geospatial agent. Geospatial information service following the OGC specifications is the ability provided by software agent, which can assemble one or more geospatial information services. Management policies including service provide policies and coordination policies are used to drive software agents to consult, analyze and co-process with each other. Geospatial knowledge base includes domain knowledge for geospatial information services.

3.2.Policy Driven Self-Management

On the one hand, the behaviors of the geospatial agent are decided by its internal cognitive model and intention. On the other hand, the behaviors of the geospatial agent are decided by its state which should be followed the external policies.

The pseudo-code of policy-driven self-management process of geospatial agent is shown below:

```
Function run
  Begin
    //Initialize knowledge base of geospatial agent
    K := K0;
//Initialize existing policy set of geospatial agent
P := P0;
//Initialize intention set of geospatial agent
    I := I0;
    while true do
      //Obtain the input information of geospatial agent
      In := Input();
 //Update the intention of geospatial agent according to //the initial and input information
      I := Fun_I(I, K, P, In);
//Determine the current policy subset followed by //geospatial agent from the existing policy set
      Ps := Fun_Ps(I, K, P);
//Obtain the actions sequence of geospatial agent //according to the policy subset
      OpSet := Fun_exePs(Ps);
//Determine whether the actions sequence has          //been executed, and whether the intention is
successful
      while not (Empty(OpSet) or Succeed(I, Ps) or Failed(I, Ps)) do
        //Obtain the first action in actions sequence
        Op1 := Fun_getOp(OpSet);
        //Update the actions sequence
        OpSet := Fun_updateOp(OpSet);
        //Geospatial agent executes the action
        Execute (Op1);
//Dynamically obtain the input information according //to the execution state
       In := Input();
//Update the policy subset according to the new input //information
        Ps := Fun_InPs(Ps, In);
//Determine whether to generate new intension or not  //according to the change of the policy subset
        If (Fun_estimate(I, Ps)) then
```

```
        //If necessary, adjust the current intention
        I := Fun_I(I, K, Ps, In);
      end-if
//Determine whether the execution action has //followed the policy and has completed the intention //of geospatial agent
If (not Fun_obey(OpSet, I, Ps)) then
//If not followed, update the current policy subset //again
Ps := Fun_Ps(I, K, P);
      end-if
    end-while
  end-while
 end
End-Function run
```

3.3.The Collaboration of Multi-geospatial Agents

In the collaborative process, management policies are used to drive geospatial agents to execute prescribed coordinated actions for maintaining the normal process. The entire process can be divided into two parts: the initial proposal process of initiator, and the feedback proposal process of the recipient. The specific process is shown in Fig. 2.

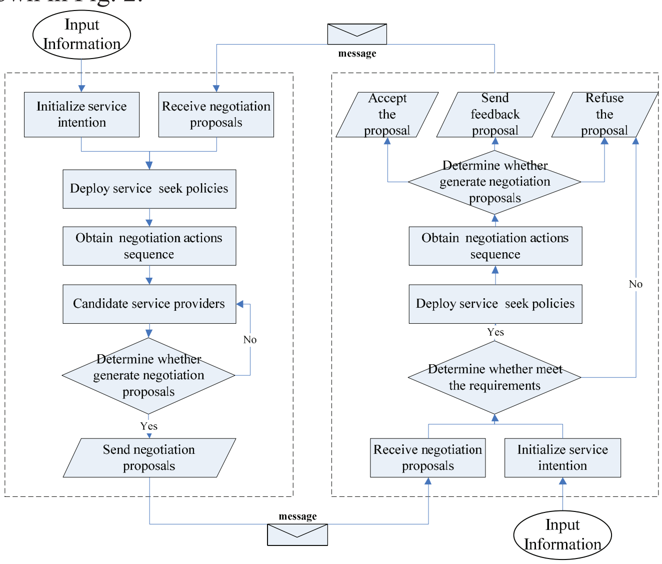

Fig. 2 Consultation process between geospatial agents

From the above figure, the initial proposal process of initiator is started by receiving input information, and by initializing geospatial knowledge base, policies set and services intention. After that, once initiator has searched enough service providers, the service negotiation phase is entered. Service seeking policies are deployed to drive initiator to obtain the negotiation actions sequence for determining whether to send proposals or not. If yes, it sends negotiation proposals to the recipient (service provider). Otherwise, it selects a new recipient in the candidate service providers and determine whether to send proposals or not again.

Similarly, the feedback proposal process of recipient is started by receiving input information and negotiation proposals of initiator, and by initializing geospatial knowledge base, policies set and services intention. The initiator determines whether to accept the feedback proposals from recipient or not. If dispute still exists between each other, then initiator sends further feedback proposals to recipient until there are no dispute exists or no agreement can be reached.

4.Agent based framework for geospatial information services composition

The multi-geospatial agents based distributed collaborative service framework for geospatial information services composition is shown in Fig. 3.

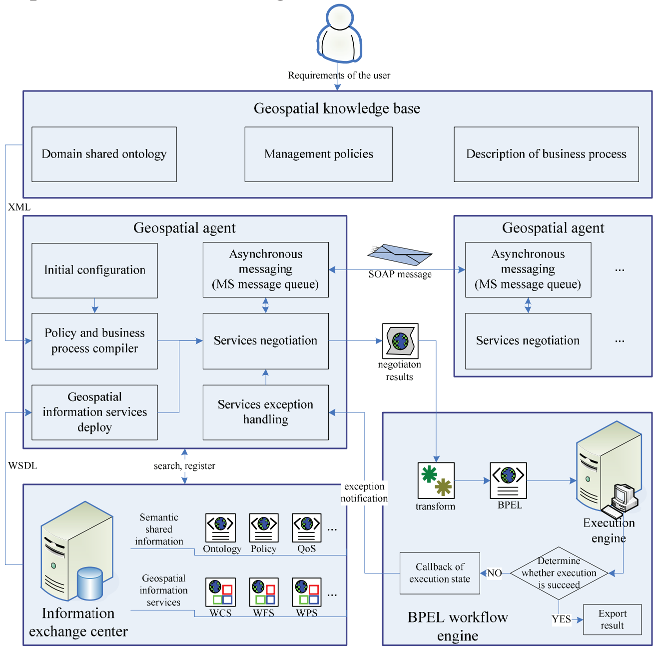

Fig. 3 Distributed collaborative framework for geospatial information services

1) Geospatial knowledge base: Geospatial knowledge base is actually common shared information, which is published and maintained in information exchange center, including domain shared ontology, management policies, and description of business process. Among them, domain shared ontology, as the semantic basis, is used to support the formal expression of other documents.

2) Geospatial agent: Geospatial agent is a software agent based program system, including initial configuration module, policy and business process compiler module, geospatial information services deploy module, asynchronous messaging module, as well as services negotiation and exception handling module. The first two modules are controlled by deploying related shared XML documents. On this basis, the services negotiation module cooperates with other agents by sending SOAP messages.

3) Information exchange center: Information exchange center mainly manages and offers related shared information to support the designers or users for choreographing the abstract service business process.

4) BPEL workflow engine: The collaborative results of multi-geospatial agents will be transformed into BPEL and then sent to BPEL workflow engine to execute when the negotiation has succeeded. If the execution fails, the workflow engine sends an exception notification to the exception handling module of geospatial agent. After that, the exception handling module determines whether to terminate or start new negotiations according to management policy.

5.Experiment and result analysis

The prototype system is composed of information exchange center, geospatial agent program system, and BPEL workflow engine (The BPELPower [15] workflow engine, developed by George Mason University, is used to execute the BPEL process).

The collaborative result between geospatial agents is an abstract process description document that confirms a set of specific participants. Correspondingly, BPEL has become the practical implementation language of web services and is supported by main business software. Therefore, services process description is transformed to BPEL services process execution.

The four parts of BPEL process expressed by the BPELPower workflow engine including bpelFile.bpel, wsdlFile.wsdl, bpel.xml and bpelInput.xml. These four documentations can be embedded in html page, and then sent to the BPELPower workflow engine through http post method. However, we can consider the BPELPower workflow engine as a web service, and invoke the getDeployment interface to deploy these four documents to it. After that, the result will be returned when process is successfully executed, or exception notified when process is failed. The result of a BPEL process is shown in Fig. 4.

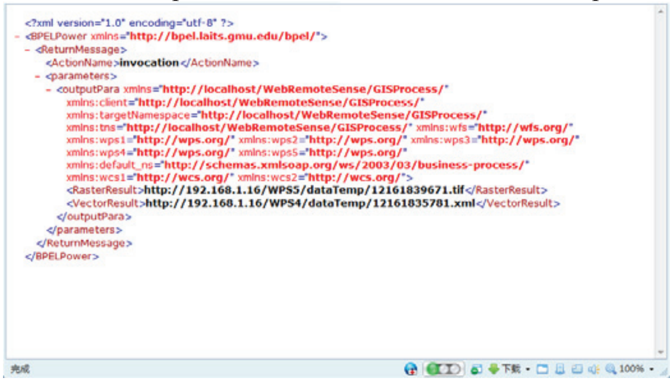

Fig. 4 Result of BPEL process

6.Conclusions

This paper proposed that policy driven and multi-geospatial agents based collaborative service framework, which combines techniques from semantic ontology, software agent, policy management and OGC specifications, is able to provide intelligent, automated distributed collaborative processing capability for geospatial information services composition. Based on policy-driven self-management, geospatial agent is capable of following the process of user's requirements to form geospatial information service chain on demand. It provides a new way for geographic information public service platform in the loose coupling integration, dynamic assembly and management of service resources.

References

[1] J. Lv, X. Ma, X. Tao, F. Xu, H. Hu, "Research and development of Internetware," *Science in China (Series E: Information Sciences)*, vol. 36, no. 10, pp. 1037-1080, 2006. (In Chinese)

[2] M. Tsou, B. Buttenfield, "A Dynamic Architecture for Distributed Geographic Information Services," *Transactions in GIS*, vol. 6, no. 4, pp. 355-381, 2002.

[3] Q. Wang, J. Wang, J. Guo, "Research on Agent-based Geospatial Information Services Composition," *Journal of Geomatics Science and Technology*, vol. 26, no. 1, pp. 60-63, 2009.

[4] S. Uttamchandani, G. Alvarez, G. Agha, “DecisionQoS: an adaptive, self-evolving QoS arbitration module for storage systems,” *In Proceedings of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY’04)*, Yorktown, NY, USA, pp. 67-76, 2004.

[5] A. Uszok, J. Bradshaw, R. Jeffers, N. Suri, P. Hayes, M. Breedy, L. Bunch, M. Johnson, S. Kulkarni, J. Lott, “KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconfliction and Enforcement,” *In Proceedings of IEEE Workshop on Policy 2003*, Como, Italy, pp. 93-98, 2003.

[6] D. Agrawal, K. Lee, J. Lobo, “Policy-based management of networked computing systems,” *Communications Magazine*, IEEE, vol. 43, no. 10, pp. 69-75, 2005.

[7] N. Damianou, “A policy framework for management of distributed systems,” London: Faculty of Engineering, University of London, 2002.

[8] J. Strassner, D. Raymer, “Implementing Next Generation Services Using Policy-Based Management and Autonomic Computing Principles,” *In Proceedings of Network Operations and Management Symposium (NOMS 2006)*, Vancouver, Canada, pp. 1-15, 2006.

[9] L. Kagal, T. Finin, A. Joshi, “A policy language for a pervasive computing environment,” *In Proceedings of the 4th IEEE International Workshop on Policy for Distributed Systems and Networks (POLICY’03)*, Los Alamitos, CA, pp. 63-76, 2003.

[10] B. Liao, J. Gao, J. Hu, J. Chen, “Ontology-based conceptual modeling of policy-driven control framework: oriented to multi-agent system for web services management,” *In Proceedings of Advanced Workshop on Content Computing (AWCC 2004)*, Berlin, Germany, pp. 346-356, 2004.

[11] B. Liao, J. Gao, J. Hu, J. Chen, “A policy-driven multi-agent system for OGSA-compliant grid control,” *In Proceedings of the International Conference on Systems, Man and Cybernetics(SMC'04)*, New York, USA, pp. 5525-5530, 2004.

[12] L. Kagal, “Rei: a policy language for the Me-Centric project, HPL-2002-270,” Palo Alto: Enterprise Systems Data Management Laboratory, HP Laboratories, 2002.

[13] N. Damianou, N. Dulay, E. Lupu, M. Sloman, “Ponder: A Language for Specifying Security and Management Policies for Distributed Systems, Version 2.3,” London: Imperial College of Science, Technology and Medicine, Department of Computing, 2000.

[14] J. Hu, B. Wang, “Overview of policy description specification research,” *Application Research of Computers*, vol. 26. No. 4, pp. 1217-1221, 2009.

[15] Center for Spatial Information Science and Systems, “BPEL Process Manager - BPELPower,” George Mason University, USA, http://data.laits.gmu.edu:9180/bpelasync/